

**Assignment Cover Sheet**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | | **Module Number and Title** | |
| Higher Diploma in Computing and Software Engineering | | CSE 4002  Fundamentals in Programming | |
| **Student Name & No.** | | **Assessor** | |
| Abdul Rahman Nazardeen CL/HDCSE/93/103 | | Vindya Karunarathne | |
| **Hand over date** | | | **Submission Date** |
| Xx.xx.xxxx | | | xx.xx.xxxx |
| **Assessment type** | **Duration/Length of**  **Assessment Type** | | **Weighting of Assessment** |
| **Coursework** | Report and Software Submission  (3000 words) | | 100% |

|  |  |
| --- | --- |
| **Learner declaration** | |
| I, Abdul Rahman Nazarden CL/HDCSZE/93/103 <name of the student and registration number>, certify that the work submitted for this assignment is my own and research sources are fully acknowledged. | |
| |  |  |  |  | | --- | --- | --- | --- | | **Marks Awarded** | | | | | First assessor | |  | | | IV marks | |  | | | Agreed grade | |  | | | Signature of the assessor |  | Date |  | |

Fundamentals in Computer Programming

Assignment

Abdul Rahman Nazardeen

ICBT CAMPUS

CL/HDCSE/93/103

17/1/2021

Author Note

This paper was produced as an assignment for Fundamentals in Networking

Contact: [abdulrahmaannazardeen@gmail.com](mailto:abdulrahmaannazardeen@gmail.com)

**Fundamentals to Computer Programming**

1. Programming methodologies

## Control Structures

## Control structures take decision on the flow of the program based on parameters. There are three types of control structures: (Deitel and Deitel, 2009)

## Decision Control Structure

## Selection Control Structure

## Repetition/Loop Control Structure

### Decision Control Structures

### Decision Control Structures are used to determine the next phase of a program based upon a criterion.

### In decision control structure the flow goes about a Boolean expression of one or more to determine the next phase, eventually deciding “true or false” after proceeding to next phase of the program. Example: If Statements. (Deitel and Deitel, 2009)

### If Statement

## 1) If Statements

If statements execute one or more statements when a condition is followed. If the testing of that condition is TRUE, the statement gets executed. But if it’s FALSE, then nothing happens.

A flow chart representation

Start

Condition

Statement

End

### Example: if condition could be used to check whether a component true or false in this example lets figure out whether 20 is greater than 18 using if condition.

### If (20>18)

### {

### cout<< “20 is greater than 18”;

### }

### Similar syntax could be used to test variables etc.

### If statement can execute statements if the tested condition is TRUE therefore “If else” is used to provide a statement for a statement with false condition.

Start

Condition

True False

End

Statement

End

### So following the previous example 20>18

### We could write something like

### If (20>18)

### {

### cout<< “20 is greater”;

### }

### Else

### {

### cout<<”18 was lesser” ;

### }

### This could be elongated using “Else IF” and can be used maybe on a grading scale, “A”, “B”, “C”, “D”, “E” and “F”. we could provide

### Example:

### int score= 77

### if(score>=90)

### {

### cout << “ Grade is A”;

### }

### else if (score>=80)

### {

### cout << “Grade is B”;}

### else if(score>=70)

### {

### cout << “Grade is C”;}

### else if(score>=60)

### {

### cout << “Grade is D”;}

### else if(score>=50)

### {

### cout << “Grade is E”;}

### else

### {

### cout << “Grade is F”;

### }

### Selection Control Structures

This control structure is used when providing answer to a specific question. Example to be used are programs that could have many problems and the statement is shown based on option selected.so switch and case statement are used. (Deitel and Deitel, 2009)

Typical syntax of this function is in c++:

Switch (expression)

{

Case constant1:

// code executed if correct;

Break;

Case constant 2:

//code executed if correct;

Break;

Default:

//executed if correct;

}

Expression

Condition

yes

statement

no

yes

statement

Condition

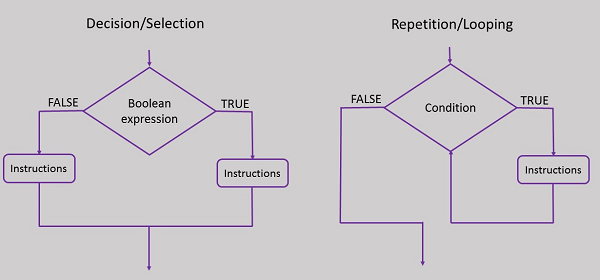
no

Default statement

switch terminates

**Repetition / Loop Control Structures**

Repetition control structures is used when statements are required to be repeated many times.

There are three forms of loops used while loop, do while loop, and for loop. (Deitel and Deitel, 2009)

The form of loop to be used could pre-determined maybe based off the previous flow chart or pseudo code used to draft the program or based on repetition planning to implement.

As represented in the above diagram the repetition loop keeps on repeating as long as the condition is true to be produce the output and once false it moves on or the iteration stops.

While loop

While loop is a loop that initiates the iteration by receiving the condition at the begin of the code and keeps on iterating as long as the condition is true and terminates when its false. (Deitel and Deitel, 2009)

Code syntax:

while (condition) {

// body of the loop

}

Example of the while loop for iteration of the variable i from 1 to 5 would look something like:

#include <iostream>

using namespace std;

int main()

{

int i =1;

while (i<=5)

{

cout << i << endl;

++i;

}

return 0;

}

This iteration starts of with defining the sequence i<=5 understanding that the sequence should be less than 5. Inside the while loop output function cout is used to show the variable content, after ++i is used to show that i=i+1 understanding what to be done by the variable.

Do While Loop

This loop begins by running the loop first afterwards evaluating the condition of the loop.

If the condition is true the loop is executed again.

The process continues till the condition is false hence terminating. (Simple control structures - C++ Tutorials, 2020)

Example in c++ for the previoous example in while would look like

do{

cout<<i<<endl;

++i;}

while(i<=5);

return 0;

For Loop

For Loop is generally used when the number of iteration is known in contrast to while and do.

For loop begins off initialization where the variable is defined.

Condition is provided if its true the command is executed once false the its terminated.

Afterwards the update or the procedure is provided. (Simple control structures - C++ Tutorials, 2020)

For loop in c++ syntax is:

for(initialization; condition: update)

{

// body

Example for the previous example would look like

For (i=1; i<=5; i++)

{cout<<i<<endl;}

Modularization

Modular technique is used since programming can get complex hence it would get hard to write, test and implement and overall big program. One error can jeopardize the entire program. A small tweak would have to look through arising difficult situation in programming.

Therefore, modularization is used to overcome by breaking the solution into smaller parts known as modules.

Breaking down the program code to smaller segments for easier development, maintenance and modification also implementation is called modularization.

Usually we do this (in c++ atleast) by using function calling these functions in the main function (Busbee, 2021)

Modular programming offers these advantages −

* Enables faster development as each module can be developed in parallel
* Modules can be re-used
* As each module is to be tested independently, testing is faster and more robust
* Debugging and maintenance of the whole program easier
* Modules are smaller and have lower level of complexity so they are easy to understand

## Identifying the Modules

Identifying modules in a software is a mind boggling task because there cannot be one correct way of doing so. Here are some pointers to identifying modules −(Busbee, 2021)

* If data is the most important element of the system, create modules that handle related data.
* If service provided by the system is diverse, break down the system into functional modules.
* If all else fails, break down the system into logical modules as per your understanding of the system during requirement gathering phase.

For coding, each module has to be again broken down into smaller modules for ease of programming. This can again be done using the three tips shared above, combined with specific programming rules. For example, for an object oriented programming language like C++ and Java, each class with its data and methods could form a single module. (Busbee, 2021)

## Step-by-Step Solution

To implement the modules, process flow of each module must be described in step by step fashion. The step by step solution can be developed using **algorithms** or **pseudocodes**. Providing step by step solution offers these advantages −(Busbee, 2021)

* Anyone reading the solution can understand both problem and solution.
* It is equally understandable by programmers and non-programmers.
* During coding each statement simply needs to be converted to a program statement.
* It can be part of documentation and assist in program maintenance.
* Micro-level details like identifier names, operations required, etc. get worked out automatically

Example

#include<iostream>

using namespace std;

void option();

int main()

{

option();

}

void options()

{

cout << “ You can call functions to the main function”;

}

As you can see you can call the function into the main function hence making large process simple by simply calling them these type of function is called Specific Task function.

Program control function on the other hand just aid to break the code easing the flow of code and giving a better out look and reduces mistakes.

2)

**System Requirement Specification**

* User login. When the user logins in with username and password the application should validate the user and directs to the main screen. If the user is invalid then an error message should pop up to him. (Screen 0 and Screen 1)
* Logout.(Screen 3)
* Exit (Screen 0)
* Main Menu (Screen 3)
* Help (Screen 2)
* Manage available Meal kits (Screen 4): Administrator should be able to select desired meal kit and select the variety available and he/she selection should be stored in a text file with the price point for later extraction. (Screen 7, 8, 9, 10, 11, 12)
* View available Meal kits- (Screen 5) User should be able to see the desired product in option 1 (Screen 13) and in option 2 (Screen 14) user should be able to see his selection of cart and their price point to be processed in option 2.
* Search specific Meal kit (Screen 6). – User should be able to search for a meal kit

User is the administrator: username: ‘AbdulRahman’ and password: ‘abdul123’

Screen 0

Option?

2

Screen 2

1

Screen 1

Correct

No

Yes

N Y

Screen 3

4

Option?

Screen 6

Screen 5

3

2

1

Screen 4

7

Option?

1

Screen 7

6

2

Screen 12

5

Screen 8

3 4

Screen 11

Screen 14

Screen 13

Option?

Screen 10

Screen 9

Software Application (C++)

Screen 0

1. User Login
2. Help
3. Exit

Enter Option (1. Login /2.Help /3.Exit)

Screen 1

**Login**

Enter username 5 attempts remaining:

Enter password:

Confirm (Yes/No)

Screen 2

**Help**

Help - Enter 1 to login and follow through the required procedure

Contact 0728091317 or email: abdulrahmaannazardeen@gmail.com for further clarification

Screen 3

**Main Menu**

1. Manage available Meal Kits
2. View available Meal Kits
3. Search specific Meal Kit
4. Logout

Enter the option (1/2/3/4)

Screen 4

**Manage available meal kit**

1. Fried rice
2. Pizza
3. Lasagna
4. Burger
5. Snacks
6. Promotions
7. Go back to main menu

Screen 5

**View Meal Kits**

1 If you want to see available Meal Kits press

2. If you want to see your Add to cart items press

3. Go back to main menu

Screen 6

**Search Meal Kits**

Type the meal kit you want to search:

Screen 7:

Fried Rice:

1. Chicken fried rice,
2. Beef fried rice,
3. Vegetable fried rice
4. Go back

Screen 8:

Pizza:

1. Beef pepporoni pizza,
2. Beef Devilled Chicken Pizza,
3. Cheese Pizza
4. Tandoori Pizza
5. Go back

Screen 9:

Lasagna:

1. Beef Lasagna,
2. Cheese Lasagna,
3. Chicken Lasagna
4. Go back

Screen 10:

Burger:

1. Beef Burger,
2. Cheese Burger,
3. Chicken Burger
4. Go back

Screen 11:

Snacks:

1. French Fries,
2. Salads,
3. Go back

Screen 12:

Promotions:

1. Big meal kit

2. Week promotions buddle

3. Go back

Screen 13:

View Meal Kits- Show Meal Kits:

1. Based on the content of text document

Screen 14:

View Meal Kits- Added to Cart Meal Kits:

1. Based on the content of text document
2. - Code attached as notepad and cpp file in zip.

T.P.O

4.

**Test cases**

1. Welcome

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| Greets: Welcome.  Shows options:  1.User Login  2.Help  3.Exit | Screen 0:  Needs to show options that are functioning, greets the user and allows user to navigate to screen 1 and 2 or else Exit the program.  If invalid option number entered shows an error invalid input and loop. | Tested and functioning well |

1. User Login

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| User Login:  Need to enter user name  After password | Screen 1:  Need to request username and password.  Needs to verify the user name is AbdulRahman or not and password: abdul123 if not give 5 more chances to enter the correct user name and password if still wrong eject from the program. | Tested and functioning well |

1. Help

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| Help option:  Displays the procedure if want further help or clarification. | Screen 2:  `Display the help message suggesting ways to call or email. | Tested and functioning well |

1. Main Menu

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| Main Menu:  After logged in should display:  1.Manage Meal Kits  2.View Meal Kits  3.Search Meal Kits  4. Log out | Screen 3:  Should display functioning options and request for the following option number thereby dragging the user to Screen 4, 5, 6 and ability to logout from the main menu and go back to home.  Also clear any add to cart meal kits in the text file.  Input should be valid if not Input invalid message should pop up and it should loop | Tested and functioning well |

1. Manage Meal Kits

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| Show the types of food category available and any promotions and the option to go back to main menu  1.Fried rice  2.Pizza  3.Lasagna  4.Burger  5.Snacks  6.Promotions  7.Go back to main menu | Screen 4:  Loop the array to display the various meal kits available.  Allow the user to go back to main menu.  Open a text file and store the meal kits available.  Open another text file as add to cart and begin to store what meal kits the user selects or prefers  Input should be valid if not Input invalid message should pop up and it should loop | Tested and functioning well |

1. View Meal Kits

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| 1.Show the type of Meal Kits available at the moment  2.Show the Meal Kits added to the cart to process  3. Go back to Main menu | Screen 5:  1.Responsible to show the available meal kits in the text document stored in manage meal kit so showing uptodate meal kits available.  2.Should loop back to the same screen  3. Should show all the meal kits added to the cart by the user preference and their prices that are selected from manage meal kit. Extract information from text file stored in manage meal kits as addtocart text file  4. Should be able to go back to main menu  Input should be valid if not Input invalid message should pop up and it should loop | Tested and functioning well |

1. Search Meal Kits

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| Search Meal Kit:  Should be able to search the text file stored in manage meal kit for its meal kits and show here | Screen 6:  User should enter what they want to search  1.Expected to show Meal kit found when discovered in the managed meal kit text file.  2. Expected to show “Not found” when not found in the text file. | Tested and functioning well  1.Tested and functioning well  2.Not working Just goes back to main menu when not discovered in the file. |

1. Log out

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| Log out : should log out from the main menu and reiterate the login process | Should go back to main menu and clear the add to cart.  Should not be able to access main menu without providing username and password again, | Tested and functioning well |

1. Manage Meal Kit – 1) Fried Rice

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| Fried Rice:   1. Chicken fried rice, 2. Beef fried rice, 3. Vegetable fried rice 4. Go back   Should show the food type and price | Screen 7:  Should be able to select the preferred choice or order both or all the food and the price and the order should be recorded in a text file  Should be able to go and select other categories and they must save in the file too  Input should be valid if not Input invalid message should pop up and it should loop | Tested and functioning well |

1. Manage Meal Kit – 2) Pizza

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| Pizza:   1. Beef pepporoni pizza, 2. Beef Devilled Chicken Pizza, 3. Cheese Pizza 4. Tandoori Pizza 5. Go back   Should show the food type and price | Screen 8:  Should be able to select the preferred choice or order both or all the food and the price and the order should be recorded in a text file  Should be able to go and select other categories and they must save in the file too  Input should be valid if not Input invalid message should pop up and it should loop | Tested and functioning well |

1. Manage Meal Kit – 3) Lasagna

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| Lasagna:   1. Beef Lasagna, 2. Cheese Lasagna, 3. Chicken Lasagna 4. Go back   Should show the food type and price | Screen 9:  Should be able to select the preferred choice or order both or all the food and the price and the order should be recorded in a text file  Should be able to go and select other categories and they must save in the file too  Input should be valid if not Input invalid message should pop up and it should loop | Tested and functioning well |

1. Manage Meal Kit – 4) Burger

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| Burger:   1. Beef Burger, 2. Cheese Burger, 3. Chicken Burger 4. Go back   Should show the food type and price | Screen 10:  Should be able to select the preferred choice or order both or all the food and the price and the order should be recorded in a text file  Should be able to go and select other categories and they must save in the file too  Input should be valid if not Input invalid message should pop up and it should loop | Tested and functioning well |

1. Manage Meal Kit – 5) Snacks

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| Snacks:   1. French Fries, 2. Salads, 3. Go back   Should show the food type and price | Screen 11:  Should be able to select the preferred choice or order both or all the food and the price and the order should be recorded in a text file  Should be able to go and select other categories and they must save in the file too  Input should be valid if not Input invalid message should pop up and it should loop | Tested and functioning well |

1. Manage Meal Kit – 6) Promotions

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| Promotions:  1. Big meal kit  2. Week promotions buddle  3. Go back  Should show the food type and price | Screen 12:  Should be able to select the preferred choice or order both or all the food and the price and the order should be recorded in a text file  Should be able to go and select other categories and they must save in the file too  Input should be valid if not Input invalid message should pop up and it should loop | Tested and functioning well |

1. View Meal Kits – 1) Show Meal Kits

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| Show Meal Kits:  Should show available Meal Kits.  Go back  Should be able to go back | Screen 13:  Should extract text from the file in manage meal kit and show in this section.  Should be able to go back and still be in the program  Input should be valid if not Input invalid message should pop up and it should loop | Tested and functioning well |

1. View Meal Kits – 2) Show Added to Cart Meal Kits

|  |  |  |
| --- | --- | --- |
| Use case | Expected | Actual |
| Show Add to Cart Meal Kits:  Should show available the Added to Cart Meal Kits.  Go back  Should be able to go back | Screen 14:  Should extract text from the file in added to cart meal kit and show in this section along with prices and with varieties selected.  Should be able to go back and still be in the program  Input should be valid if not Input invalid message should pop up and it should loop | Tested and functioning well  Had to clear the file in Main menu so added an ofstream without amend command hence overwriting and erasing.  (Command included in this section ofstream Myfile(“abc.txt”, ios::app) which amended the commands in main menu use without amend to clear the file. |

**Reference**

* Cplusplus.com. 2020. *Simple Control Structures - C++ Tutorials*. [online] Available at: <http://www.cplusplus.com/doc/oldtutorial/control/> [Accessed 17 January 2021].
* Deitel, H. and Deitel, P., 2009. *4.2 Control Structures | C++ For Programmers: Control Statements: Part 1 | Informit*. [online] Informit.com. Available at: <https://www.informit.com/articles/article.aspx?p=1321841&seqNum=2#:~:text=C%2B%2B%20has%20only%20three%20kinds,while).> [Accessed 17 January 2021].
* Busbee, K., 2021. *2.3: Modularization And C++ Program Layout*. [online] Engineering LibreTexts. Available at: <https://eng.libretexts.org/Bookshelves/Computer\_Science/Book3A\_Programming\_Fundamentals\_-\_A\_Modular\_Structured\_Approach\_using\_C\_\_\_(Busbee)/02%3A\_Introduction\_to\_Programming/2.03%3A\_Modularization\_and\_C\_Program\_Layout> [Accessed 17 January 2021].